Paper ID #19548

Using Programming and Simulation as a Research Tool in a Graduate Capital Budgeting Course

Dr. James D Burns, Purdue Univeristy

Jim Burns, Ph.D. Assistant Professor, Department of Technology Leadership & Innovation Jim joined the faculty at Purdue Polytechnic in 2015 after completing a Ph.D. in Industrial Engineering from Western Michigan University, and has more than 10 years industry experience in the manufacturing sector in a variety of roles including process engineering, operations management, and technical sales. His area of expertise centers on applying OR/MS and Simulation techniques to Supply Chain & Operations Management problems, and has also conducted research in the areas of Human Factors and Work Design for evaluating time and motion efficiencies of operations. Jim also holds an undergraduate IE degree and a Six Sigma Greenbelt.

Dr. Bob E. White, Western Michigan University

Bob White has a Ph.D. in Engineering Valuation from Iowa State University (1980). He is currently a professor of Industrial and Entrepreneurial Engineering & Engineering Management at Western Michigan University. His interests include entrepreneurial engineering, engineering economy, capital budgeting, and operations control. Dr. White is actively involved as a consultant to industry having been involved with numerous companies. Dr. White has over 20 journal publications and more than 50 articles in conference proceedings. Some of Dr. White's journal publications include The Engineering Economist, Computers and Industrial Engineering, The International Journal of Modeling and Simulation, and The International Journal of Production Research. His professional affiliations include or have included IIE, ASEE, and SME.

Using Programming and Simulation as a Research Tool in a Graduate Capital Budgeting Course

Abstract: Effectively incorporating research elements into the classroom continues to be an important aspect of graduate engineering education. In this paper, we present an overview of a semester-long research project for a graduate course in Capital Budgeting. At the core of the project is a student-developed Capital Budgeting simulation model. This model is used both as a tool for examining the effectiveness of existing capital project ranking methods such as IRR, NPV, and Payback, and also as a research platform for testing new ranking and selection methods. The project consists of three phases. The first phase is comprised of individual work where students design and program a basic Capital Budgeting simulation model using Microsoft Excel and Visual Basic for Applications (VBA). During the second phase, students transition into small self-selected teams to validate and then expand the simulation to include more financial elements. This work is performed over the first two-thirds of the semester. The third phase involves teams investigating various project ranking and selection techniques available in the peer-reviewed literature and then using the simulation model to perform experiments comparing those techniques. Students also design and examine new capital project selection techniques that can be applied either to the general Capital Budgeting problem or to specific real-world situations. The project concludes with each team presenting the results of their work to the class.

Grounded in project-based learning principles, this project provides an additional way for students to apply Engineering Economy concepts throughout the course. We outline current instructional techniques, discuss assignments given within each phase, and review the evaluation criteria for each phase and assignment. We also discuss particular instructional challenges we have observed that stem from the varied programming skills, comfort levels with Engineering Economy concepts, industry experiences, academic concentrations, and nationalities we find in the classroom. Finally, we present a preliminary comparison of outcomes related to the quality of simulation models observed after specific instructional techniques related to programming with VBA were modified.

Introduction

In this paper we discuss our approach to administering a semester-long project in a graduate course that focuses on Engineering Economy, Capital Budgeting, and cost analysis topics. The project involves the development of a simulation model using Microsoft® Excel® and Visual Basic for Applications® (VBA) for the purpose of investigating the classic investment problem of maximizing wealth in a capital-rationed environment. Simulation is a tool that is widely used in financial analysis and is commonly used in finance-related courses[1, 2], and this particular project has its origins in work that one of the authors did with Professor Gerald W. Smith at Iowa State University[3]. It has been modified and enhanced throughout several years of use in the graduate programs at Western Michigan University.

The project is a year-by-year simulation of the traditional project ranking/selecting, capital recovery, and reinvestment processes that characterizes the Capital Budgeting activities found in many organizations. The objectives of the project are to reinforce financial concepts learned in the course, to further emphasize to students the value of simulation for analysis and research, and to provide a platform for students to carry out experiments for course-related research.

The use of VBA to construct the simulation and our approach to instruction are interrelated. The use of programming to set up and conduct experiments is widely used in research, and we believe that by leveraging students' relative comfort with Excel as they become accustomed to VBA the burden of learning a new programming language is eased. It is our experience that the vast majority of students are familiar with Excel, but relatively few are familiar with VBA. Additionally, although Monte Carlo simulation is widely used in finance-related courses, it is not particularly well-suited for this project. The year-by-year approach, the number of projects ranked, and the need for flexibility in extracting and tabulating intermediate data would render a model based purely in Excel unwieldy due to the volume of iterative spreadsheet calculations.

One strength of this project is that the simulation model itself is developed in parallel with the introduction of financial concepts. Students must understand these concepts well in order for them to be effectively integrated into the model. Although challenging for students early on in the semester, we believe there is great value in students working with the two new subjects (Capital Budgeting and VBA) over time and within the same context (the simulation). Development work is distributed throughout the first two-thirds of the semester to create a sense of flow in relation to the topics of the course. After development, students use the remainder of the semester to examine peer-reviewed literature in order to find other methods for ranking and selecting projects, to design new or novel approaches to project selection, and to modify the simulation model to better reflect real-world operating environments. Students then use the simulation model to perform experiments and test hypotheses related to research objectives they define.

Of course, there are challenges to this approach, not the least of which is the relatively short time students have to learn and apply a new programming language. As with most instructional challenges, finding the right approach for the situation is more desirable than attempting to balance between extreme options (leaving students to learn on their own versus providing a complete tutorial on VBA programming). Our intent here is to introduce the approach taken for this course in order to highlight those aspects we believe are most critical to success.

Instruction

Overview

The project begins during the first week of class and concludes during the last meeting. The project is administered in three phases, with the model-building activities performed during Phase 1 and Phase 2. Phase 1 of the project consists of two individual assignments (A1and A2) that when completed will produce, in rough form, the working simulation model. Phase 2 requires students to self-select teams and to continue developing the model by way of three assignments (A3 through A5). The first assignment of Phase 2 (A3) involves correcting, simplifying, and validating the functionality of the model. The second assignment (A4) involves adding the ability to borrow capital and creating an income statement for the purpose of incorporating depreciation, interest, and taxes. Once these first two assignments are completed, teams are given a third assignment (A5) consisting of investigating a small research-based problem using the model.

Phase 3 consists of a single assignment (A6) in which teams are given the task of reviewing academic literature in order to better understand both historical and contemporary perspectives on Capital Budgeting. The findings of the literature review are then used to test different ranking methods that have been either drawn from the literature or developed by the team. Students are also encouraged to add elements of realism, risk, and variability to the models. Table 1 provides information pertaining to the sequence of assignments, classroom instruction time dedicated to each assignment, duration of the assignment, and the percent value of each assignment in relation to the overall project score. Specific requirements for each assignment are discussed in the following sections.

The evaluation criteria varies by assignment and is based on the individual requirements of the work and their relation to the broader objectives for the course. During Phase 1, when students are just learning to program, the emphasis is placed on effort that is evidenced by an attempt to

Table 1 – Assignments for the Capital Budgeting Simulation Project

Phase	Assign.	Grade Weight	Evaluation Emphasis	Instruction Time	Assignment Duration
1	A1	4%	- Apparent effort in learning VBA and progress toward requirements. - Emphasis on generating correct cash flows, the presence of programming loops and a table of results.	1 Hour	1-2 Weeks
	A2	5%	- Model is complete and code utilizes loops, variables, and parameters to control the operation of the model. - Output matches expected results. - The work demonstrates that the student understands basic programming concepts and the underlying concepts of the simulation.	45 Minutes	1-2 Weeks
2	A3	5%	- Model is correct as measured by simulation output, cash flow calculations are correct, ranking/buying logic is correct. - Excel’s Solver employed to select projects by maximizing NPV.	45 Minutes	2 Weeks
	A4	7%	- Model is correct as measured by simulation output. - Logic for borrowing is correct and Income Statement calculations (Earnings Per Share, Net Income) are correct.	45 Minutes	2 Weeks
	A5	4%	- Correctness of the model as measured by the characteristics of projects purchased and simulation output. - Presentation of the experimental procedures used to estimate the cut-off ROR as instructed. - Comparison of simulation output when MARR = cut-off ROR.	30 Minutes	1 Week
3	A6	75%	- Use of literature to identify and test different ranking methods. - Development of unique approaches to project selection. - Incorporating elements of realism into project selection. - Presenting the results of the experiments in a compelling manner. - Quality and professionalism of the written report.	45 Minutes	4-6 Weeks

meet the requirements. During Phase 2, the emphasis transitions from effort to accuracy, and the evaluation is based on the conformance of the model's output to target values produced by a validated reference model. The goal of Phase 2 is to develop a valid model so that Phase 3 can begin with a functional research platform. The work from Phase 3 carries most of the weight in terms of the overall grade for the project, and is based on the quality of the research, experimental design, analysis, written report, and a formal presentation made to the class.

Phase 1

The underlying objectives of A1 are for students to become familiar with VBA programming and to conceptualize the logic of the model. The specific requirements of the first individual assignment are shown in Figure 1. Students are given instruction on the basic logic sequence for the simulation and on best practices for using VBA in conjunction with Excel. The basic logic sequence is described in Figure 2. This sample material represents a three-year simulation that students may use as a template for model construction, for checking the underlying logic of their program, and for validating the calculations for annual cash flows, Net Present Value (NPV), and Payback, which are all dependent upon the randomly-generated projects.

The layout of the simulation in Excel is an important component of successful development, and strategies for constructing the simulation are discussed during the introduction to A1. The example shown in Figure 2 depicts one possible layout, and represents the layout we recommend to students. The primary advantage of the layout is that the simulation model can essentially be contained to a single worksheet, allowing cash flows and ranking parameters to be viewed (and validated) easily. This layout also allows the use of Excel's data sort functions for ranking

A1 Requirements

1. Create a Capital Budgeting simulation model using VBA and Excel to evaluate the performance of three project ranking methods (Payback, IRR, and NPV). The simulation will operate on a year-by-year basis for 5 years where some initial amount of capital is used to invest in projects at Time-0, and any remaining capital plus financial returns are used to reinvest in subsequent years. After 5 years of investing, sum any remaining capital with any uncollected returns to determine overall the total amount of wealth possessed, called here Net Wealth.
2. Projects will be created using probability distributions for their various parameters. These and other important simulation parameters are:

Initial Capital	\$600,000	Project First Cost	~ U(\$40,000, \$280,000)
# of years to invest:	5	Project Life	~ U(2 years, 10 years)
# of projects available the first year:	25	Project IRR	~ U(5%, 15%)
Yearly increase in projects:	5	Discount Rate (MARR)	15%

3. Perform 30 simulation replications for each of the three ranking methods. Compare the resulting Net Wealth for each ranking method using a statistical comparison.

A2 Requirements

1. Expand the simulation to include additional ranking methods: NPV/FC, AW, AW/FC, Random
2. Expand the simulation to operate for 10 years.
3. Create a table to compile the First Cost, Life, and IRR for each individual project purchased, and for all ranking methods and replications.
4. Include an area in the Excel portion of the model to allow all parameters from Assignment 1 to be changed, and ensure that the simulation model responds accordingly.

Figure 1 – Requirements for Phase 1 Assignments

Step 1: Beginning of Year 1, there is enough initial capital ($50,000) to buy 2 of the ranked projects.

50,000 – 382,579 = 117,421

Step 2: At the end of Year 1, cash from the projects are collected (186,634) and combined with capital left over.

117, 421 + 186,634 = 304,055

Step 3: Beginning of Year 2, there is enough capital (304,055) to buy 1 of the ranked projects available.

304,055 – 219,105 = 84,950

Step 4: At the end of Year 2, collect cash from projects (287,698) and combine with capital left over.

84,950 + 287,698 = 372,648

Step 5: Beginning of Year 3, there is enough capital (372,648) to buy 3 of the ranked projects available.

372,648 – 331,935 = 40,713

Step 6: At the end of Year 3 (also the end of the buying), tally all uncollected cash from projects & left over capital.

40,713 + 2,0570880 = 2,098,593

	A	B	C	D	E	F	G	H	I	J	K	L	Q	R	S
1	Cash Flows Collected Each Year →						186,634	287,698	419,716	419,716	287,005	....	45,015	0	0
2	Project Statistics & Information						Projected Cash Flows from Projects								
3	Buy?	Year	Payback	NPV	IRR	First Cost	EOY 1	EOY 2	EOY 3	EOY 4	EOY 5	....	EOY 10	EOY 11	EOY 12
4	1	1	1.85	133,473	40.0%	(245,412)	132,710	132,710	132,710	132,710					
5	1	1	2.54	120,133	37.0%	(137,167)	53,923	53,923	53,923	53,923	53,923	....			
6		1	4.66	14,196	17.0%	(183,617)	39,415	39,415	39,415	39,415	39,415	....	39,415		
7		1	5.65	(21,594)	12.0%	(193,220)	34,197	34,197	34,197	34,197	34,197	....	34,197		
8		1	4.49	(24,177)	9.0%	(154,618)	34,467	34,467	34,467	34,467	34,467	....			
9	1	2	2.17	163,371	40.0%	(219,105)		101,064	101,064	101,064	101,064	....			
10		2	2.29	68,912	37.0%	(106,055)		46,233	46,233	46,233	46,233	....			
11		2	2.51	132,596	35.0%	(201,155)		80,221	80,221	80,221	80,221	....			
12		2	3.24	35,484	26.0%	(92,246)		28,465	28,465	28,465	28,465	....			
13		2	3.89	(1,769)	14.0%	(66,023)		16,978	16,978	16,978	16,978	....			
14		2	5.22	(7,792)	14.0%	(205,958)		39,485	39,485	39,485	39,485	....	39,485	39,485	
15		2	5.97	(51,372)	7.0%	(206,713)		34,618	34,618	34,618	34,618	....			
16	1	3	2.48	90,139	37.0%	(111,859)			45,015	45,015	45,015	....	45,015		
17	1	3	2.40	48,358	37.0%	(66,209)			27,537	27,537	27,537	....			
18	1	3	2.59	71,177	31.0%	(153,867)			59,465	59,465	59,465	....			
19		3	2.92	118,085	30.0%	(221,016)			75,569	75,569	75,569	....	75,569		
20		3	3.67	64,459	23.0%	(215,538)			58,680	58,680	58,680	....	58,680	58,680	
21		3	3.42	53,818	22.0%	(246,761)			72,247	72,247	72,247	....			
22		3	2.17	10,297	18.0%	(205,482)			94,506	94,506	94,506	....			
23		3	4.34	6,174	16.0%	(186,584)			42,956	42,956	42,956	....	42,956		
24		3	5.33	(18,303)	10.0%	(115,204)			21,594	21,594	21,594	....	21,594		

Figure 2 – Simulation Logic Sequence

projects (e.g., greatest to least according to IRR), thereby eliminating the need for students to write sorting algoritms early in the project. When a binary variable for selecting projects is used as shown in Column A of the Figure 2, the layout becomes well-suited to integrating Excel's native solver platform, which is required in A3. The layout is also convenient for transferring the information into a programming array to speed up calculations as students' programming skills mature. The purpose of the recommended layout is not to constrain students (there are many approaches that can be taken), but to prevent poor choices which result in large, overly-complex models that span several worksheets.

The output for the simulation at this phase is the *Net Wealth* that results from investing. In the example shown in Figure 2, *initial capital* of $50,000 is used to select projects for a three-year period. Corresponding parameters from Figure 1 include *# of projects available the first year*, which in this case is five, and the *yearly increase* in projects, which is two. Individual projects are generated using similar parameters for project life and project first cost as those shown in Figure 1. In this example projects are ranked according to IRR and the resulting *Net Wealth* is $2,098,593. This represents one replication for one ranking method (IRR). *Net Wealth* values are tabulated for each ranking method for 30 replications, and students are able to compare their results to target values created from a validated reference model. An example of simulation target values are shown in Table 2.

Table 2 – Target Values for Net Wealth for A1

	Ranking Method		
(*n* = 30)	Payback	NPV	IRR
Mean	2,059,506	4,717,299	4,437,962
Standard Deviation	263,703	443,530	416,882

For programming, we emphasize simplicity over sophistication and encourage students to leverage the visual power of Excel to build the model. Using VBA to perform calculations (e.g., calculating returns using the PMT function) and placing numerical values in the cells instead of spreadsheet formulas is often very counterintuitive for students. Other key programming topics and VBA syntaxes are discussed. We demonstrate the use of the most essential programming elements such as *Do* and *For* loops, the use of variables, how to reference cells using VBA's *Cells, Range, With* and *Offset* objects/statements. Although we list an optional VBA reference text[4], we have not found textbook references to be helpful in the context of this project. We provide an Excel file containing code that demonstrates all of the essential syntax and programming constructions needed to complete the first several assignments (and in reality, to complete the project). We generally discourage students from relying on code from recorded macros to form any central part of the model, but discuss how macros can be useful in learning about the capabilities and syntax of VBA.

The principal objectives for A2 are to improve understanding of VBA and to create a working simulation model that operates as intended and produces the expected results for Net Wealth. Students are asked to expand the work begun in A1 by adding additional ranking methods, operating the simulation for 10 years, and tabulating information on each project selected. Employing programming loops, variables, parameters to control the operation of the simulation, and functions to collect intermediate data are required at this time. The additional ranking methods include Annual Worth (AW), AW divided by First Cost (AW/FC), NPV divided by First Cost (NPV/FC), and random selection to serve as a basis for comparison.

Upon completion of Phase 1 students self-select teams that will work together for the remainder of the semester, usually consisting of 3-4 members. There are multiple considerations when forming groups for project-based coursework, and we tend to agree with some of the arguments in favor of self-selected groups[5] for this project. However, in order to overcome some of the natural problems that accompany team-based projects[6] we stipulate that each student demonstrate a minimum level of competence with VBA and demonstrate a minimum conceptual understanding of the goal of the project before being allowed to join a team. Students who do not are required to either continue developing the simulation model individually until they demonstrate competence or to take a programming quiz. Very few fail to demonstrate minimum competency by the completion of A2.

Phase 2

The general requirements for each assignment in Phase 2 are shown in Figure 3. The objectives for the first team assignment (A3) are to have teams agree on a modeling approach, to correct any remaining logic or calculation errors, and to add the functionality necessary to solve the well-known Capital Budgeting optimization problem using Excel's native solver. Each team's task is to choose a modeling strategy and use their collective abilities to develop a valid simulation model. Teams usually opt to refine the "best" model created by one of the members during Phase 1, but occasionally a new model is developed from scratch. At this point in the series of assignments each simulation model is generally functional and complete, and the evaluation now includes assessing the validity of the output as measured by a set of target values obtained from a reference model. The optimization problem is formulated as a linear Binary Integer Program (BIP), and the instructional emphasis in this phase centers on the implementation of Solver to solve the BIP formulation as described by the course textbook[7]. It is at this time the rationale for the recommended layout becomes more evident to students.

The objective of A4 is to incorporate additional financial concepts into the simulation. Requirements include adding the ability to borrow capital using a prescribed debt ratio and incorporating the effects of interest expense, depreciation expense, and taxes by way of an Income Statement. This is perhaps the most challenging assignment for students, in-part because they are just being introduced to these concepts in the classroom but also because it is necessary to accommodate depreciation expense, interest payments, and principal payments on a project-by-project and year-by-year basis. With the introduction of the income statement, students are required to calculate both the NPV of remaining capital and future after-tax cash flows (ATCF)

A3 Requirements

1. Make changes and corrections to a simulation model in order to meet all objectives set forth in A1 and A2, and validate the output versus target values.
2. Use the Classic Capital Budgeting Model in conjunction with Excel's Solver add-in to select projects on a year-by-year basis, and track Net Wealth for using this method for 30 replications.

A4 Requirements

1. Add the ability to borrow capital given the assumptions listed below:
    - Use a 20% Debt Ratio. Assume $480,000 of equity capital is available initially, and that up to $120,000 can be borrowed (Total Capital $600,000).
    - Assume an 8% interest rate on borrowed capital.
    - Assume debt is repaid using equal payments over the life of the project.
2. Add Straight Line depreciation to fully depreciate the First Cost of each project purchased over its life.
3. Add an Income Statement to the model for the purposes of calculating Earnings Per Share and After Tax Cash Flow given the assumptions listed below:
    - Tax rate is 40%.
4. Add output tables to the simulation model for tracking NPV @ Year-10 and Earnings Per Share.
5. Include an area in the Excel portion of the model to allow all parameters from Assignment 1 to be changed, and ensure that the simulation model responds accordingly and is able to achieve the target values provided.

A5 Requirements

1. Run several simulation replications ranking by IRR only. Tabulate the IRR of the last project purchased in each year. Calculate the average of these IRR values to estimate the cut-off ROR.
2. Run several simulation replications for all ranking methods where the MARR is indexed up from 15%. Plot the average Life and First Cost of all projects purchased versus the MARR, and find the value for MARR where the average Life and First Cost begin to converge on the graph in order to estimate cut-off ROR.
3. Run 30 simulation replications where MARR = cut-off ROR. Prepare charts that analyze Net Wealth and NPV @ Year-10 and perform a statistical comparison on these variables. What are your observations?

Figure 3 – Requirements for Phase 2 Assignments

after the investing period of the simulation has ended (Year-10), as well as the annual Earnings per Share from the Income Statement. These outputs represent both an operational profitability measurement and the traditional Engineering Economy metric of ATCF. The instructions are to assume that shares are issued at $10 per share to finance the initial equity capital investment of $480,000. In the same manner as for Net Wealth, target values for these new metrics are provided so that students may use them for model validation.

The instructional materials provided for this phase are drawn from a reference simulation model and include examples of the Income Statement plus information regarding strategies for tracking depreciation, debt payments, and interest payments. Certain assumptions for borrowing and depreciation are made at this time. Straight Line depreciation is used to recover the entire first cost of a project over its life (zero salvage value). Regarding borrowed capital, the stipulation is that projects are purchased with a capital ratio equal to the assumed debt ratio and that borrowed capital is repaid in equal annual payments over the life of the project. However unrealistic from a real-world operations perspective these assumptions may be, they are necessary to simplify the coding within VBA, and are subsequently relaxed in Phase 3.

The final assignment in Phase 2 (A5) introduces students to concepts related to experimentation and research using the simulation model. In A5 students are asked to find the cut-off rate of return (CoROR) using two estimation procedures (See Figure 3) and to use this estimated rate in place of the given discount rate. We assume here that the discount rate and the Minimum Attractive Rate of Return (MARR) are one and the same. In the context of the capital budgeting simulation the CoROR can be described as the IRR of the last project purchased when projects are ranked according to IRR. A comparison of results between simulations using the assumed MARR and CoROR is performed to observe the effect on financial performance and the characteristics of the projects selected. The intent here is to illustrate how the discount rate, which is often taken for granted in financial calculations, can have significant influence over returns and the characteristics of the projects selected. As an example, Figure 4 displays average first costs for all projects purchased when the MARR is varied for three ranking/selection methods. By constructing similar graphics, students will be able to observe the effect that changing the discount rate has on a variety of outputs, including Net Wealth, NPV at Year-10,

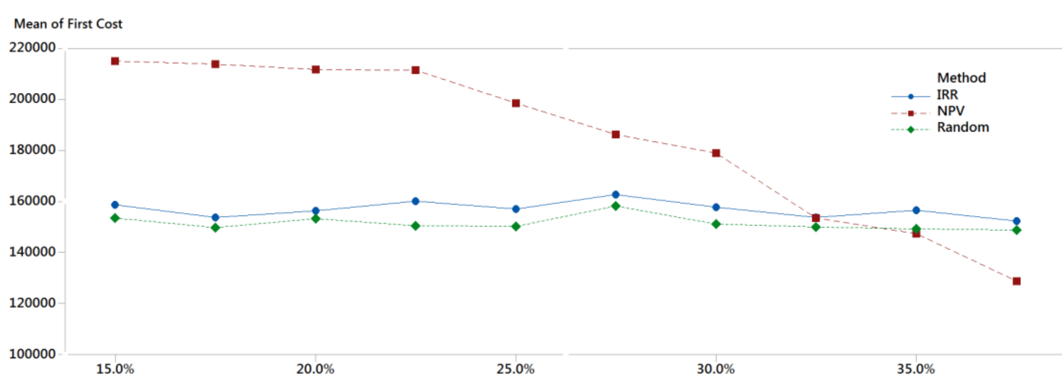

Figure 4 – Project First Cost by Simulation MARR

average project life, and average project first cost. If their model is constructed properly, students will observe that the convergence of many of these outputs occur near the CoROR, thereby demonstrating the value of exploratory research.

Phase 3

Upon completion of Phase 2, student teams will have constructed a functional simulation model that can be used as a research tool for investigating different project ranking/selection methods. Students should at this time also have a basic understanding of Engineering Economy principles and spreadsheet modeling techniques and should also be quite familiar with VBA as it relates to the construction of their simulation model. The final assignment (A6) involves reviewing the academic literature to identify alternate selection/ranking methods, and enhancing their models by adding elements of realism. Students are instructed to use their improved model to perform experiments in order to test hypotheses they develop as a team.

By this point in the semester, students are very likely conditioned to believe that higher financial returns are always better and sometimes see A6 as a mandate to find ways to improve the financial metrics. However, our emphasis here is on adding realism to the simulation improvements, finding different or new ways to select projects, crafting well-designed experiments, and performing thorough analyses. Realism, which usually includes adding elements of risk and variability, will nearly always degrade financial performance. Although this assignment is intended to be open-ended, there are some common characteristics that emerge in students' final results. The most common changes students make to the simulation are adding variability to cash flows and adding optimistic-pessimistic-likely risk parameters to projects. The most common ranking method additions include the well-known MIRR, Discounted Payback, and Economic Value Added methods for selecting projects.

For A6, the instructional emphasis centers on how students may find articles in the peer-reviewed literature using the university's library system and on discussing some possible

A6 Requirements

1. Identify and test capital project ranking methods found in the academic literature.
2. Improve the simulation model by adding elements of realism. These can be from the literature, textbooks, your experience in industry, or other sources (e.g., various risk factors).
3. Develop and test new methods for ranking and selecting projects.
4. Perform experiments to compare performance of the various ranking methods relative to each other.

Suggestions for Simulation Model Realism and Research

- Add elements of risk/variability of project performance, and incorporate those into project selection criteria.
- Tailor the simulation to represent conditions from a specific operating environment or industry sector.
- Add considerations for risks associated with future interest rates or capital availability.
- Ranking methods that include risk assessments typically use parameters to quantify that risk (e.g., probabilistic assessments of pessimistic or optimistic scenarios). It is usually a good idea to perform sensitivity analysis on such parameters.
- For any project selection method, be sure that it does not rely on the impossible, such as having *a priori* knowledge of future events or actual financial performance.
- Begin your research with a financial concept or question your team finds interesting.

Figure 5 – Requirements for Phase 3 Assignment & Points of Emphasis

strategies for adding realism including those listed above. The specific requirements given to student teams for Phase 3, along with points of emphasis passed on to students during the introduction to A6, are shown in Figure 5.

Instructional Emphasis, Observations, and Evaluation of Recent Changes

Over time we have come to believe that the primary challenges to administering the simulation project stem from the initial learning of programming with VBA. Because the course focuses on Capital Budgeting it is natural that general financial concepts become refined over time. Moreover, student understanding of simulation grows over time because other assignments given throughout the course make use of an Excel-based Monte Carlo simulation package. However, it is not our experience that student comfort with VBA programming naturally becomes refined to the same degree within the context of this project. Computer programs tend to become very large and "entangled" over time and can be very difficult to "detangle" quickly. We believe that emphasizing fundamental techniques to simplify and streamline the program early on and gearing these techniques toward those inexperienced with VBA can help reduce the likelihood that students' programs will become very large and cumbersome to modify.

One example of a programming technique that is important to learn is the combining of variables and programming loops with various simulation parameters, such as the *# of years to invest*, *# of projects available the first year*, and *yearly increase in projects*. When treated as variables and combined with algebraic expressions for an arithmetic sequence, the parameters mentioned can be leveraged to fully control the creation of the randomly-generated projects. They can also be leveraged to identify locations of interest within the spreadsheet (or within a programming array) for project ranking and selection. We illustrate this technique in class with the information shown in Figure 6. This is perhaps the most detailed code example we provide. It is intended not only to assist students in learning how to use variables and expressions, but also to show how VBA's Offset property is used and to further illustrate our recommended layout for the simulation. Programming techniques such as those illustrated in Figure 6 can dramatically reduce the number of lines of code needed for the simulation, but these are techniques a novice programmer will not likely be aware of without specific instruction. We believe the most essential techniques for this project can be summarized by the points listed in Figure 7.

The instructional emphasis as described in this paper has long been a part of the course but has been applied in a more focused fashion in the two most recent semesters through the use of newly-developed instructional materials. While the volume of instructional materials precludes fully describing them in this paper, and their efficacy will be determined over a longer period of time, it is possible to make a few initial observations. Figure 8 presents a comparison between key evaluation characteristics observed in models from the most recent semesters without (*Before*) and with (*After*) the new instructional materials and emphasis.

This comparison is comprised of models submitted by students for A4. A total of seven models were available for comparison from the semester prior to the instructional changes, and nine models were available from the semester after the changes. Each model was examined for four

It is possible to use the simulation's parameters to create expressions which simplify coding. This is especially useful when writing loops for the capital budgeting simulation.

In this example, loops and mathematical expressions are used to populate a table.

Parameters & Variables

noYears → # of Buying Years
noProj → # of Projects in Year 1
projGrowth → Project Growth/Year

Variables

thisYear → The Current Year
thisProj → The Current Project
indxProj → Indexing variable

The code below demonstrates a simple loop to write random numbers between 50 and 100 into a spreadsheet starting in Cell A2. The same structure could populate several individual arrays, or one multi-dimension array. Assume these random numbers are first costs for projects.

```
Sub Program1()
  noYears = 3
  noProj = 5
  projGrowth = 2
  indxProj = 1

  With Range("A2")
    For thisyear = 1 To noYears
      For thisProj = 1 To noProj + (thisYear - 1) * projGrowth
        .Offset(indxProj - 1, 0).Value = indxProj
        .Offset(indxProj - 1, 1).Value = thisYear
        .Offset(indxProj - 1, 2).Value = thisProj
        .Offset(indxProj - 1, 3).Value = Rnd * (100 - 50) + 50
        indxProj = indxProj + 1
      Next
    Next
  End With
End Sub
```

	A	B	C	D
1	Index	Year	Project	Cost
2	1	1	1	85
3	2	1	2	77
4	3	1	3	79
5	4	1	4	64
6	5	1	5	65
7	6	2	1	89
8	7	2	2	51
9	8	2	3	88
10	9	2	4	91
11	10	2	5	85
12	11	2	6	52
13	12	2	7	71
14	13	3	1	93
15	14	3	2	90
16	15	3	3	69
17	16	3	4	98
18	17	3	5	94
19	18	3	6	53
20	19	3	7	97
21	20	3	8	68
22	21	3	9	76

Parameters & Variable Values

noYears = 3
noProj = 5
projGrowth = 2

Figure 6 – Instructional Material, Programming Using Variables and Expressions

criteria: 1) The use of nested loops in the programming, 2) the use of variables/code to control all simulation functions according to the parameters provided in A1, 3) the correctness of Income Statement calculations, and 4) the successful integration of debt capital into the simulation. These criteria represent a continuum of both time and difficulty in relation to the construction of the simulation model. Nested loops are an essential component and are easy to implement during A1. Parametric control is more difficult, but is something that should be implemented by the completion of A3. As mentioned previously, adding an Income Statement is often very challenging for students, and the use of Debt Capital even more so. These are often not fully implemented until after A5 is complete.

- Utilize VBA to perform calculations rather placing formulas in the spreadsheet (e.g., PMT, NPV, or calculating Payback). Our recommendation is that no spreadsheet formulas are used, except for those necessary for optimization using Solver.
- Utilize programming loops, specifically For and Do loops, starting with the first assignment.
- Utilize variables and parameters to control as many aspects of the simulation as possible beginning with the first assignments (e.g., number of replications, random distribution characteristics, the minimum attractive rate of return).
- Practice extracting and compiling intermediate results into a spreadsheet database for analysis early on in the development process.
- Choose a layout for the spreadsheet that facilitates visual validation of calculations (similar to that shown in Figure 2) when "stepping through" the program.
- Choose spreadsheet layout that allows for easy implementation of Excel's native Solver platform.

Figure 7 – Points of Instructional Emphasis for Initial VBA Instruction

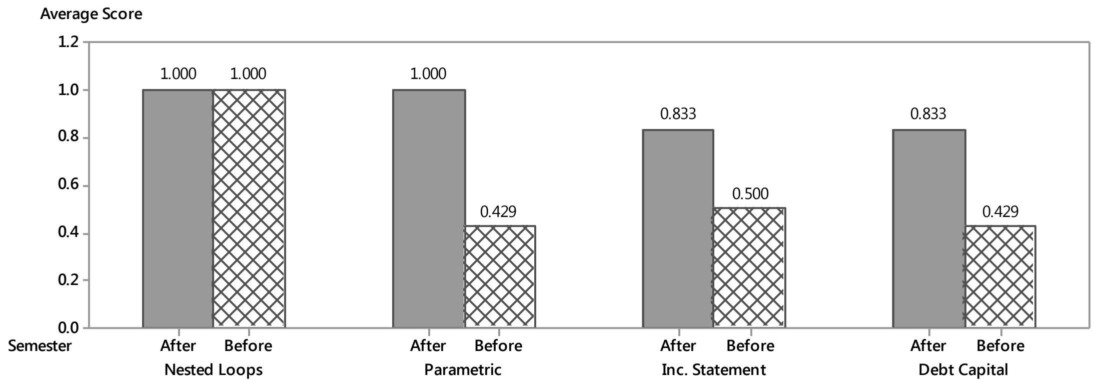

Figure 8 – Simulation Construction Before and After Instruction Changes

A scoring system ranging from 0.00 to 1.00 was applied to this comparison. A score of 1.00 indicated that the criteria was met, a score of 0.75 indicated that the criteria was met but small errors were found, and a score of 0.50 indicated an attempt was made to meet the criteria but significant errors exist. A score below 0.50 indicated that no attempt was made or that very significant errors were made. Recall that the goal upon completion of A4 is not to have a fully valid model, but rather to have a solid model framework that can be corrected easily and validated upon completion of A5. The rationale for evaluating the characteristics here is that early instruction on programming best-practices should result in fully parameterized models after A3 is submitted. If accomplished, it is more likely that students will successfully integrate the Income Statement and debt capital into the simulation.

The results of the comparison seem to indicate a positive shift. The most significant improvement is that all the models developed after the instructional changes were fully controlled by parameters. There was also a positive shift in that student teams were more successful in regards to incorporating borrowed capital. Additionally, an overall improvement was observed in the accuracy of Income Statement calculations. (Note: Any calculation errors associated with the Income Statement are usually addressed by the end of the project.) Student teams both before and after the implementation of instructional changes were successful in using nested loops.

Discussion

The course in which this project is administered is a requirement for multiple graduate programs, including the M.S. and Ph.D. in Industrial Engineering and M.S. in Engineering Management. Students come to the course from a wide array of professional disciplines, undergraduate majors, and cultural backgrounds, and they have varying degrees of comfort both with financial topics and computer programming. There is a healthy mix of full-time and part-time students. Attempting to make a general statement about how any of these factors influence student success, except perhaps for a background in programming, would be a mistake. Our motivating interest is simply to better understand what kinds of instructional methods and support are most effective in helping students develop a working simulation model that can be used for course-related research.

While the changes to instructional materials give specific guidance on how to program the most complex aspects of the simulation and a complete model could be developed from the sections of sample code provided, they are not intended to provide a full tutorial on how to develop the model. We leave it to students to use the materials as they see fit. Some students seem to prefer to forgo using reference material altogether and complete A1 and A2 using a combination of macro-generated code and very rudimentary programming structures. Once project teams are formed, the actual programming work is usually performed by one or two ambitious individuals who enjoy the work while the remaining group members contribute in other ways. This seems to work well for students, and anecdotally there have been relatively few complaints regarding the distribution of the workload in the project.

However, despite the modest improvements observed in the initial comparison here and a handful of positive comments from students, it is worth asking what else can be done to improve the process so that students are even more likely to begin Phase 3 with a working, valid model that can be modified further. Our belief is that adding additional reference material without first considering whether or not it will help the larger goal of in-class research is counterproductive. Past attempts to provide nearly-complete sections of code or Excel templates to get the coding process started have not substantially improved the final results. It can also be argued that without an in-depth knowledge of how the model is constructed, modifying it to add realism or to test other ranking methods would be extraordinarily difficult. Without question, there is an element of this project that simply requires a focused effort to learn enough about VBA programming to complete the job.

Our intention with the recent changes was to provide more precise requirements for what we believe are the most critical aspects of the work and to provide instruction and reference material to support those objectives. The ultimate responsibility to make use of those resources still lies with the students and teams. To help determine whether or not this approach has practical value, we intend to perform more in-depth analysis that includes examining additional assignments (e.g., A2 and A6) and additional programming aspects (e.g., lines of code, amount of macro-generated code, accuracy of output), while expanding the analysis to include additional semesters.